

SPECIFICATION

Docket No. 10001579-1

TO ALL WHOM it may concern:

BE IT KNOWN that we, Joe Hunt, a citizen of the United States, and Julio Garcia, a citizen of the United States, have invented new and useful improvements in

**Runtime Configurable Caching For Component Factories**

of which the following is a specification:

09372035-060101  
T07090-5302650

1 BACKGROUND OF THE INVENTION:

3 1. TECHNICAL FIELD

5 The present invention relates generally to object-oriented development of  
6 software applications and more specifically to object caching strategies in  
7 object-oriented applications.

9 2. BACKGROUND OF THE INVENTION

11 Caching strategies are often a part of the software development process  
12 when application speed and overhead are important considerations. A  
13 caching strategy allows a software application to retrieve frequently used  
14 results more quickly by interacting with special memory locations with faster  
15 access time than general purpose computer memory. Caching is often done  
16 using specialized hardware or at the operating system level. This type of  
17 caching is not easily accessible to an object-oriented application. Applications  
18 that need higher level caching build their own type-specific caches. The  
19 process of building type-specific caches is laborious and can significantly  
20 increase the application development overhead. Adding caching for new  
21 object types can be difficult, since each new type of object has specific  
22 memory and storage requirements that must be met for optimal performance.  
23 These criteria make it difficult to obtain a caching strategy that is consistent  
24 across many different object types. Caches are an integral part of the  
25 application and tuning the cache for optimal performance can have a  
26 significant coding impact across the application.

28 A mechanism that allows object-oriented applications to configure caches for  
29 the objects that they use would allow application performance to improve.







1 After configuring the cache object, the application may assign one or more  
2 cache factory objects to the cache encapsulated by the cache object. The  
3 cache factory objects inherit from a factory object, and interact directly with  
4 the application and the cache object to locate, add, remove, or otherwise  
5 manipulate objects from the cache.  
6

09/07/2009 06:04:04

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21

## BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the claims. The invention itself, however, as well as a preferred mode of use, and further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

**Figure 1** shows an example of how an application, factory objects, and cache interact, according to the present invention.

**Figure 2** shows a sequence diagram of the object interactions, according to the present invention.

**Figure 3** is a flowchart illustrating how an application interacts with an object factory to obtain an object, according to the present invention.

**Figure 4** is an interface hierarchy diagram, according to the present invention.

1

## 2 DESCRIPTION OF THE INVENTION

3

4 While this invention is susceptible of embodiment in many different forms,  
5 there is shown in the drawings and will herein be described in detail specific  
6 embodiments, with the understanding that the present disclosure is to be  
7 considered as an example of the principles of the invention and not intended  
8 to limit the invention to the specific embodiments shown and described. In the  
9 description below, like reference numerals are used to describe the same,  
10 similar or corresponding parts in the several views of the drawing.

11

12 The present invention discloses a method and structure for integration of run-  
13 time configurable caches with object factories in an object-oriented  
14 application. According to this method, an object-oriented application  
15 containing one or more object factories may use one or more caches  
16 associated with the one or more object factories in order to provide fast  
17 access to objects associated with the application. According to the structure  
18 of the present invention, objects are provided which encapsulate the cache  
19 from the object-oriented application. These objects contain methods that  
20 allow the object-oriented application to manipulate the objects contained in  
21 the cache.

22

23 Referring now to **Figure 1**, an exemplary architecture 100 of an object-  
24 oriented application 110, three object factories, two caches, and a database  
25 150 is shown. For the purposes of this example, the database 150 contains  
26 three objects: R, G, and B. Application 110 interacts with object factory 122,  
27 object factory 125, and object factory 127, in order to use the R, G, and B  
28 objects. The two caches in **Figure 1** are used by the three object factories to  
29 store the R, G, and B objects. Cache 130, which interacts with object factory



1 122, stores the B object. Cache 140, which interacts with object factory 125  
2 and object factory 127, stores the R and G objects. Note that one cache, in  
3 this case cache 140, can be associated with more than one object factory.  
4 When application 110 requests one of the R, G or B objects, the object factory  
5 associated with these objects can retrieve the object directly from the  
6 appropriate cache, thereby saving the time required for a database access.  
7 For example, if application requests the B object, then object factory 122  
8 interacts with cache 130 to retrieve the B object and return it to application  
9 110. Note that in the preferred embodiment, an object is returned to  
10 application 110 by reference, although a copy of the object may be returned  
11 without departing from the spirit and scope of the invention.

12  
13 Referring now to **Figure 2**, a sequence diagram showing the interaction  
14 between the application, factory object, and cache objects is shown.  
15 Application 110 first sends a create message 235 to cache factory object 215.  
16 Cache factory object 215 creates cache object 225. Cache object 225 may  
17 then be configured by application 110. Application 110 sets the Class Cache  
18 ID by sending a setClassCacheID message 240 to cache object 225.  
19 Application 110 then sends a setMaxSize message 245 to cache object 225 in  
20 order to set the size of the created cache object. These commands may be  
21 used to configure the cache object for use with various types of objects. Once  
22 application 110 has configured cache object 225, application 110 sends a  
23 setCache message 250 to a factory object 220, thereby associating factory  
24 object 220 with cache object 225. This association allows objects created by  
25 factory object 220 to use cache object 225. Factory object 220 can interact  
26 with cache object 225 to add objects to the cache, remove objects from the  
27 cache, retrieve messages from the cache, or perform other similar types of  
28 operations. After application 110 sends the setCache message 250 to factory  
29 object 220, application 110 may directly request an object contained in cache



1 indicates how the cache handles objects that do not match the corresponding  
2 object contained in database 150. When an object becomes stale, it may be  
3 removed immediately or only marked as stale, depending upon the value of  
4 the cache object type obtained using the getType method. A cache item  
5 object 460, which derives from cache configuration object 440, contains  
6 methods to add an object to the cache, remove an object from the cache, or  
7 find an object located in the cache.

8  
9 The factory object 220 contains methods to create a cache object, get a  
10 classID, and get a database connection instance. Factory object 220  
11 inherently couples the created cache object to database 150 through the use  
12 of the get database connection instance method. Cacheable factory object  
13 450 derives from factory object 420 and contains methods to set and get the  
14 cache associated with factory object 220. It should be noted that in the  
15 preferred embodiment of the present invention, other classes may be used to  
16 provide enhanced cache management strategy without departing from the  
17 spirit and scope of the present invention. It is further noted that the database  
18 object 150 may interact with Microsoft® ODBC™, Oracle™, Sybase™, or any  
19 database element that has similar operating characteristics.

20  
21 Thus, in the present invention, a single cache may be coupled to one or more  
22 object factories, and therefore coupled to one or more databases, through the  
23 use of cache objects, cache factory objects and unique identifiers assigned to  
24 each object in the cache. The cache objects and cacheable factory objects  
25 interact with the object-oriented application in order to provide a transparent  
26 interface between the application, the database and the cache. That is, the  
27 application does not need to deal with how objects are retrieved from the  
28 cache or stored in the cache. Also, the use of cache objects and cache  
29 factory objects encapsulate the cache so that during run-time operation the

1 cache may be manipulated. One aspect of this manipulation is that objects  
2 may be added, removed or located within the cache, and these objects can be  
3 organized so that the correspondence between the accuracy of these objects  
4 relative to the database they came from is established.

6 While the invention has been particularly shown and described with reference  
7 to a preferred embodiment, it will be understood by those skilled in the art that  
8 various changes in form and detail may be made therein without departing  
9 from the spirit and scope of the invention.

Attorney Docket Number: 10001579-1